Information and Computation **152**, 205–235 (1999)
Article ID inco.1999.2797, available online at http://www.idealibrary.com on IDEAL®

Logic with Equality: Partisan Corroboration and Shifted Pairing*

Yuri Gurevich†

Microsoft Research, One Microsoft Way, Redmond, Washington 98052-6399
E-mail: gurevich@microsoft.com

and

Margus Veanes‡

Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany
E-mail: veanes@mpi-sb.mpg.de

Herbrand's theorem plays a fundamental role in automated theorem proving methods based on tableaux. The crucial step in procedures based on such methods can be described as the *corroboration* problem or the *Herbrand skeleton* problem, where, given a positive integer m, called *multiplicity*, and a quantifier free formula, one seeks a valid disjunction of m instantiations of that formula. In the presence of *equality*, which is the case in this paper, this problem was recently shown to be undecidable. The main contributions of this paper are two theorems. The first, the *Partisan Corroboration Theorem*, relates corroboration problems with different multiplicities. The second, the *Shifted Pairing Theorem*, is a finite tree automata formalization of a technique for proving undecidability results through direct encodings of valid Turing machine computations. These theorems are used in the paper to explain and sharpen several recent undecidability results related to the *corroboration* problem, the *simultaneous rigid E-unification* problem and the *prenex fragment of intuitionistic logic with equality*.

1. INTRODUCTION

We study classical first-order logic with equality but without any other relation symbols. The letters φ and ψ are reserved for quantifier-free formulas. The *signature* of a syntactic object X (a term, a set of terms, a formula) is the collection of function symbols in X augmented, in the case when X contains no constants, with a constant c. The language of X is the language of the signature of X.

* Preliminary versions of this paper have appeared as: UPMAIL Technical Report 138, Uppsala University, 1997, and Research Report MPI-I-98-2-014, Max-Planck-Institut für Informatik, 1998.

† Partially supported by the NSF Grant CCR-95-04375.

‡ Corresponding author.

0890-5401/99 $30.00

Any syntactic object is *ground* if it contains no variables. A substitution is *ground* if its range is ground, and it is said to be in a given language if the terms in its range are in that language. A set of substitutions is *ground* if each member is ground.

Given a positive integer m, a set of m ground substitutions $\{\theta_1, ..., \theta_m\}$ is an *m-corroborator* for φ if the disjunction $\varphi\theta_1 \vee \cdots \vee \varphi\theta_m$ is provable. A ground substitution θ *corroborates* φ if $\{\theta\}$ 1-corroborates φ; such a θ is called a *corroborator* for φ.

One popular form of the classical Herbrand theorem (e.g., Herbrand, 1972) is this:

> *An existential formula* $\exists \mathbf{x}\varphi(\mathbf{x})$ *is provable if and only if there exists a positive integer* m *and* m*-corroborator for* φ *in the language of* φ.

The minimal appropriate number m will be called the *minimum multiplicity for* φ. The minimum multiplicity for a formula may exceed one. Here is a formula for which the minimum multiplicity is two, suggested by Erik Palmgren in a different but similar context; we use "$\approx$" for the formal equality sign:

$$(c \approx c_0 \Rightarrow x \approx c_1) \wedge (c \approx c_1 \Rightarrow x \approx c_0)$$

The Herbrand theorem plays a fundamental role in automated theorem proving methods known as the *rigid variable methods* (Voronkov, 1997). We can identify the following procedure underlying such methods. Let $\exists \mathbf{x}\varphi(\mathbf{x})$ be a closed formula that we wish to prove.

THE PRINCIPAL PROCEDURE OF RIGID VARIABLE METHODS

Step I. Choose a positive integer m.

Step II. Check if there exists an m-corroborator for φ.

Step III. If Step II succeeds then $\exists \mathbf{x}\varphi(\mathbf{x})$ is provable, otherwise increase m and return to Step II.

The kernel of the principal procedure is of course Step II or

THE CORROBORATION PROBLEM.

Instance: A quantifier free formula φ and a positive integer m.

Question: Is the minimum multiplicity for φ bounded by m?

Corroboration for a fixed m is called *m-corroboration*. A detailed discussion of corroboration and related problems is given by Degtyarev, Gurevich, and Voronkov (1996). It is important to us here that corroboration is intimately related to existential intuitionistic provability and simultaneous rigid E-unification (Gallier, Raatz, and Snyder, 1987). The first of these problems is easy to formulate:

THE EXISTENTIAL INTUITIONISTIC PROVABILITY PROBLEM.

Instance: An existential formula $\exists \mathbf{x}\varphi(\mathbf{x})$.

Question: Is the formula provable in intuitionistic logic with equality?

The second requires auxiliary definitions. A *rigid equation* is expression $E \vdash^{\mathrm{r}} e$, where E is a finite set of equations and e is an equation. A ground substitution θ *solves* a rigid equation $E \vdash^{\mathrm{r}} e$ if $e\theta$ is a logical consequence of $E\theta$. A system (that is a finite set) of rigid equations is *solvable* if there is one substitution that solves all rigid equations in the system.

THE SIMULTANEOUS RIGID E-UNIFICATION PROBLEM (SREU).

Instance: A system of rigid equations.

Question: Is the system solvable?

The SREU problem has an interesting history (e.g., Degtyarev, Gurevich, and Voronkov, 1996). Several false decidability claims have been published until, finally, Degtyarev and Voronkov (1995) proved SREU to be undecidable. Moreover, Plaisted (1995) has shown that the fragment of SREU with ground left-hand sides is already undecidable (the *left-hand side* of a rigid equation $E \vdash^{\mathrm{r}} e$ is E).

It is easy to see that SREU is essentially a special case of one-corroboration for Horn formulas. Hence, the result of Degtyarev and Voronkov shows that corroboration is undecidable already in this very special case. Voronkov (1997) has suggested the following generalization of the corroboration problem. Let f be a function that assigns a positive integer to every pair (k, φ), where k is a positive integer and φ a formula in our logic. Moreover, it is assumed that $k < l$ implies that $f(k, \varphi) \leqslant f(l, \varphi)$. Such a function is called a *strategy* for multiplicity. The intended meaning of the first argument of a strategy is the number of times that Step II of the principal procedure has been executed.

THE CORROBORATION PROBLEM WITH STRATEGY f.

Instance: A quantifier free formula φ and a positive integer k.

Question: Is the minimum multiplicity for φ bounded by $f(k, \varphi)$?

Corroboration with a strategy that does not depend on it arguments, i.e., takes a constant value m for all arguments, is simply m-corroboration. Voda and Komara (1995) have proved that, for each positive integer m, the m-corroboration problem is undecidable. One important conclusion for automated theorem proving, drawn by Voda and Komara, is that there is no m for which one can effectively determine whether m bounds the minimum multiplicity for a given formula. The proof of Voda and Komara is very technical, and we wondered if there is a way to derive their result from the Degtyarev–Voronkov theorem. It turns out that indeed there is such a way.

In order to formulate our results, we need to recall a few definitions and give definitions of our own. Recall that a *Horn clause* is a disjunction of negated atomic formulas and at most one nonnegated atomic formula; a Horn clause is often represented as a set of its disjuncts. Here we restrict attention to Horn clauses that contain exactly one nonnegated atom. A *Horn formula* is a conjunction of Horn clauses. Since the equality sign is the only relation symbol in our logic, every Horn clause ψ is equivalent to an implication $E \Rightarrow s \approx t$, where E is a conjunction of equalities.

We say that a collection of formulas is *constant-disjoint* if there is no constant that occurs in two or more of the given formulas. Call a Horn formula φ *guarded* if, for every variable x that occurs in φ, there exists a clause $E \Rightarrow s \approx t$ in φ, where E and s are ground and x occurs in t. Finally, call a corroborator θ of a disjunction φ *partisan* if already θ corroborates one of the disjuncts of φ. Now we are ready to formulate our first result.

PARTISAN CORROBORATION THEOREM. *Every corroborator for a disjunction of constant-disjoint guarded Horn formulas is partisan.*

This theorem is proved in Section 3. We believe it is of independent interest. It allows us an easy derivation of Voda and Komara's (1995) result from Degtyarev and Voronkov's (1995) theorem in Section 4. Moreover, we strengthen the theorem of Voda and Komara in several ways. For each m, we effectively reduce SREU to the m-corroboration problem in such a way that the positive-arity part of the signature remains unchanged. In particular, for every m, the monadic (all function symbols are of arity at most one) SREU reduces to monadic m-corroboration; this reduction is of interest because the decidability of monadic SREU is an open problem.

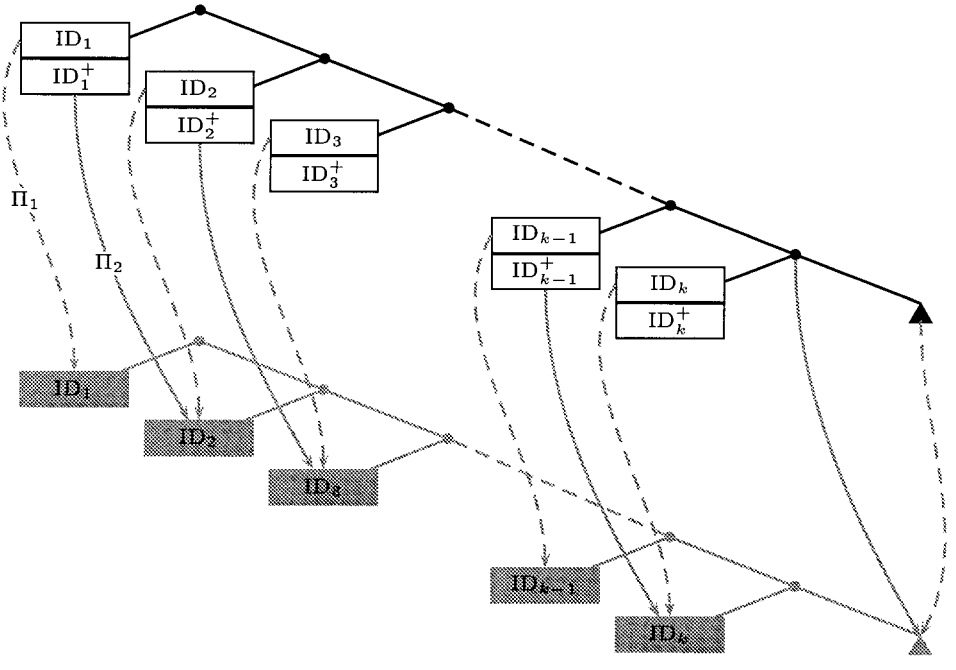

FIG. 1. Shifted pairing. Each term t recognized by A_{id} represents a sequence of *IDs* of M. Each term s recognized by A_{mv} represents a sequence of *moves* of M. If s reduces in Π_1 to t then the *first projection* of s coincides with t. Moreover, if s reduces in Π_2 to the *tail* of t, then the *second projection* of s coincides with the tail of t.

In Section 5 we use the *finite tree automata* theory to describe a powerful technique, named *shifted pairing* by Plaisted (1995), for proving undecidability results via encodings of valid Turing machine computations. The main components are two finite tree automata A_{mv}, A_{id} and two ground term rewrite systems Π_1 and Π_2 that are obtained (effectively) from a given Turing machine M and are used to check the existence of a *valid computation of* M (see Fig. 1).

SHIFTED PAIRING THEOREM. *There are two finite tree automata* A_{mv} *and* A_{id} *and two ground rewrite systems* Π_1 *and* Π_2 *such that it is undecidable whether, given a ground term* t_0, A_{mv} *recognizes a term* s *and* A_{id} *recognizes a term* t, *such that* s *reduces in* Π_1 *to* t *and* $f(t_0, s)$ *reduces in* Π_2 *to* t.

A more precise version of the theorem is stated in Section 5. The shifted pairing technique, and in particular the Shifted Pairing Theorem, that is an improvement upon (Plaisted, 1995; Veanes, 1997), has recently been applied successfully to settle several open decidability questions (Ganzinger, Jacquemard, and Veanes, 1998; Levy and Veanes, 1998; Veanes, 1997, 1998).

In Section 6, we use the Shifted Pairing Theorem to show the undecidability of a fragment of SREU with only two variables and three rigid equations with ground left-hand sides, which constitutes the currently known *least undecidable fragment* of SREU. Using this result and the Partisan Corroboration Theorem, we show for each positive integer m the undecidability of m-corroboration when each formula is a conjunction of $3m$ Horn clauses with $2m$ variables and ground negative literals.

In Section 7 we obtain some undecidability results related to the prenex fragment of intuitionistic logic with equality and proof search in intuitionistic logic with equality. Finally, in Section 8 we describe the current status of SREU and related results and mention some open problems.

2. PRELIMINARIES

We will first establish some notation and terminology. We follow Chang and Keisler (1990) regarding first-order languages and structures. For the purposes of this paper it is enough to assume that the first-order languages that we are dealing with are languages with equality and contain only function symbols and constants, so we will assume that from here on. We will in general use Σ, possibly with an index, to stand for a signature; i.e., Σ is a collection of function symbols with fixed arities. A function symbol of arity 0 is called a *constant*. We will always assume that Σ *contains at least one constant*.

2.1. Terms and Formulas

Terms and formulas are defined in the standard manner and are called Σ-*terms* and Σ-*formulas*, respectively, whenever we want be precise about the language. We refer to terms and formulas collectively as *expressions*. In the following let X be an expression or a set of expressions or a sequence of such.

We write $\Sigma(X)$ for the *signature of* X: the set of all function symbols that occur in X, $FV(X)$ for the set of all free variables in X, and $\text{Con}(X)$ for the set of all

constants in X. We write $X(x_1, x_2, ..., x_n)$ to express that $FV(X) \subseteq \{x_1, x_2, ..., x_n\}$. Let $t_1, t_2, ..., t_n$ be terms; then $X(t_1, t_2, ..., t_n)$ denotes the result of replacing each (free) occurrence of x_i in X by t_i for $1 \leqslant i \leqslant n$. By a *substitution* we mean a function from variables to terms. We will use θ to denote substitutions. We write $X\theta$ for $X(\theta(x_1), \theta(x_2), ..., \theta(x_n))$.

We say that X is *closed* or *ground* if $FV(X) = \varnothing$. By $\mathscr{T}_\Sigma$, or simply $\mathscr{T}$, we denote the set of all ground Σ-terms. A substitution is called *ground* if its range consists of ground terms.

A closed formula is called a *sentence*. Since there are no relation symbols all the atomic formulas are *equations*, i.e., of the form $t \approx s$, where t and s are terms and "$\approx$" is the formal equality sign.

Atomic formulas and negated atomic formulas are called *positive* and *negative literals*, respectively. A *clause* is a disjunction of literals. By a *Horn clause* we mean a clause with exactly one positive literal (i.e., a *strict* Horn clause). A Horn clause can be written as $E \Rightarrow s \approx t$, where E is a (possibly empty) conjunction of equations, and s and t are terms. By a *Horn formula* we understand a conjunction of Horn clauses.

2.2. First-Order Structures

First-order structures will (in general) be denoted by $\mathscr{A}$ and $\mathscr{B}$. A first-order structure in a signature Σ is called a *Σ-structure*. For $f \in \Sigma$ we write $f^{\mathscr{A}}$ for the interpretation of f in $\mathscr{A}$.

If $\mathscr{A}$ is a Σ-structure and $\Sigma' \subseteq \Sigma$ then $\mathscr{A} \restriction \Sigma'$ is the Σ'-structure that is the reduction of $\mathscr{A}$ to signature Σ'. Let $\mathscr{A}$ and $\mathscr{B}$ be Σ-structures, $\mathscr{A}$ is a *substructure* of $\mathscr{B}$, in symbols $\mathscr{A} \subseteq \mathscr{B}$, if the universe of $\mathscr{A}$ is a subset of the universe of $\mathscr{B}$, and for each n-ary $f \in \Sigma$, $f^{\mathscr{A}}$ is the restriction of $f^{\mathscr{B}}$ to the universe of $\mathscr{A}$.

For X a sentence or a set of sentences, $\mathscr{A} \models X$ means that the structure $\mathscr{A}$ is a *model of* or *satisfies* X according to Tarski's truth definition. A set of sentences is called *satisfiable* if it has a model. If X and Y are (sets of) sentences then $X \models Y$ means that Y is a *logical consequence* of X, i.e., that every model of X is a model of Y. We write $\models X$ to say that X is *valid*, i.e., true in all models.

By the *term algebra over* Σ we mean the Σ-structure $\mathscr{A}$ with domain $\mathscr{T}_\Sigma$, such that for each n-ary $f \in \Sigma$ and $t_1, ..., t_n \in \mathscr{T}_\Sigma$, $f^{\mathscr{A}}(t_1, ..., t_n) = f(t_1, ..., t_n)$. We let $\mathscr{T}_\Sigma$ also stand for the term algebra over Σ.

Let E be a set of ground equations. Define the equivalence relation $=_E$ on $\mathscr{T}$ by $s =_E t$ if and only if $E \models s \approx t$. By $\mathscr{T}_{\Sigma/E}$ (or simply $\mathscr{T}_{/E}$) we denote the quotient of $\mathscr{T}_\Sigma$ over $=_E$; i.e., for all $s, t \in \mathscr{T}$

$$\mathscr{T}_{/E} \models s \approx t \Leftrightarrow E \models s \approx t.$$

2.3. Term Rewriting

In most cases we consider a system of ground equations as a rewrite system. We will assume that the reader is familiar with basic notions regarding ground term rewrite systems (e.g., Dershowitz and Jouannaud, 1990). We will only use very elementary properties. In particular, in the next section we will use Birkhoff's (1935) completeness theorem for equational logic in the case of ground equations.

THEOREM 1 (Birkhoff). *Given a ground set of equations E and a ground equation $s \approx t$, $E \models s \approx t$, if and only if s can be reduced to t by using the equations in E as rewrite rules in both directions.*

Let R be a ground rewrite system. We write $R^{\approx}$ for the corresponding set of equations:

$$R^{\approx} = \{ s \approx t \mid s \to t \in R \}.$$

In Section 6 we will use the following property of ground canonical (or convergent) rewrite systems R (e.g., Dershowitz and Jouannaud, 1990, Section 2.4). For any two ground terms t and s, the equation $t \approx s$ follows logically from $R^{\approx}$ if and only if the normal forms of t and s in R coincide, i.e.,

$$R^{\approx} \models t \approx s \Leftrightarrow t\downarrow_R = s\downarrow_R.$$

Snyder (1989) has given a very simple but useful condition for showing that a ground rewrite system R is canonical, namely that it is *reduced*: for each rule $s \to t$ in R, s is irreducible in $R \backslash \{ s \to t \}$ and t is irreducible in R. We will use this test on several occasions to show that a ground rewrite system is canonical.

2.4. Finite Tree Automata

A finite tree automaton or TA is a quadruple (Q, Σ, R, F), where

- Q is a finite set of constants called *states*,
- Σ is a *signature* that is disjoint from Q,
- R is a set of rules of the form $f(q_1, ..., q_n) \to q$, where $f \in \Sigma$ has arity $n \geqslant 0$ and $q, q_1, ..., q_n \in Q$,
- $F \subseteq Q$ is the set of *final states.*

A TA is called *deterministic* or a DTA if there are no two different rules in it with the same left-hand side. Terms are also called *trees* and a *forest* is a set of trees. The forest *recognized* by a TA $A = (Q, \Sigma, R, F)$ is the set that is denoted by $\mathscr{L}(A)$:

$$\{ t \in \mathscr{T}_\Sigma \mid (\exists q \in F)\ t \xrightarrow{*}_R q \}.$$

A forest is *recognizable* or *regular* if it is recognized by some TA. A well-known fact is that every regular forest is recognized by a DTA. Two finite tree automata are called *constant-disjoint* if there is no constant that occurs in both of them.

EXAMPLE 2. Let $A = (\{q\}, \Sigma, R, \{q\})$ be a TA, where

$$R = \{ c \to q \mid c \text{ is a constant in } \Sigma \}$$
$$\cup \{ f(q, ..., q) \to q \mid f \text{ is a function symbol in } \Sigma \}.$$

This DTA recognizes the forest $\mathscr{T}_\Sigma$. ∎

3. PARTISAN CORROBORATION THEOREM

The following lemma is used in the Partisan Corroboration Theorem. We say that two (sets of) expressions X and Y are *constant-disjoint* if $\mathrm{Con}(X) \cap \mathrm{Con}(Y) = \varnothing$.

LEMMA 3. *Let φ_i for $i \in I$, be pairwise constant-disjoint quantifier free sentences. Then $\models \bigvee_{i \in I} \varphi_i$ implies $\models \varphi_i$ for some $i \in I$.*

Proof. For $i \in I$, let $\Sigma_i = \Sigma(\varphi_i)$ and let $\Sigma = \bigcup_i \Sigma_i$. Assume by contradiction that $\not\models \varphi_i$ for all $i \in I$. Then there is (for each $i \in I$) a Σ_i-structure $\mathscr{A}_i$ such that $\mathscr{A}_i \models \neg\varphi_i$. Without loss of generality, take the universes of all the models to be pairwise disjoint.

We now construct a Σ-structure $\mathscr{A}$ such that $\mathscr{A}_i \subseteq \mathscr{A} \restriction \Sigma_i$ for $i \in I$. First, let the universe of $\mathscr{A}$ be the union of the universes of the $\mathscr{A}_i$'s. Next, for each $i \in I$ and constant $c \in \Sigma_i$ let $c^{\mathscr{A}} = c^{\mathscr{A}_i}$. For each n-ary function symbol f in Σ define $f^{\mathscr{A}}$ as follows. For all individuals $\mathbf{a} = a_1, ..., a_n$ in $\mathscr{A}$,

$$f^{\mathscr{A}}(\mathbf{a}) = \begin{cases} f^{\mathscr{A}_i}(\mathbf{a}), & \text{if } a_1, ..., a_n \text{ are in } \mathscr{A}_i; \\ a_1, & \text{otherwise.} \end{cases}$$

It is clear that $\mathscr{A}$ is well defined because of the disjointness criteria and that $\mathscr{A}_i \subseteq \mathscr{A} \restriction \Sigma_i$ for $i \in I$. One easily establishes by induction on terms and formulas that, if $\mathscr{B} \subseteq \mathscr{A}$ then for all quantifier free sentences φ, $\mathscr{B} \models \varphi$ if and only if $\mathscr{A} \models \varphi$. Hence, $\mathscr{A} \restriction \Sigma_i \models \neg\varphi_i$ for $i \in I$, and thus, $\mathscr{A} \models \neg\varphi_i$ for $i \in I$. But this contradicts that $\models \bigvee_{i \in I} \varphi_i$. ∎

Note that Lemma 3 can be seen as a particular case of the Łoś–Tarski theorem (existential sentences are preserved under extensions).

If we drop the constant-disjointness criterion in Lemma 3, then of course the lemma is false. A simple counterexample is

$$\models c_0 \approx c_1 \vee \neg(c_0 \approx c_1).$$

We will state now some other obvious but useful lemmas. Lemma 4 is an easy corollary of Birkhoff's completeness theorem.

LEMMA 4. *Let t and s be ground terms and let E and E' be ground sets of equations such that $\mathrm{Con}(E') \cap (\mathrm{Con}(E) \cup \mathrm{Con}(s)) = \varnothing$. Then $E' \cup E \models t \approx s$ implies $E \models t \approx s$.*

Proof. Let E, E', s, and t be given and assume that $E' \cup E \models t \approx s$. By Birkhoff's (1935) completeness theorem we know that s can be rewritten to t by using $E' \cup E$ as a set of rewrite rules. So there is a sequence of terms $s_0, s_1, ..., s_{n-1}, s_n$, where $s_0 = s$, $s_n = t$, and s_i is rewritten to s_{i+1} by using some rule in $E' \cup E$ for $0 \leqslant i < n$. By induction on i (for $i \leqslant n$) it follows that $\Sigma(s_i) \subseteq \Sigma(E) \cup \Sigma(s)$ and only a rule from E can be used to rewrite s_i. The statement follows from the completeness theorem of Birkhoff. ∎

LEMMA 5. *Let t and s be ground terms and let E be a ground set of equations. Then $E \models t \approx s$ implies $\Sigma(t) \subseteq \Sigma(E) \cup \Sigma(s)$.*

Proof. Take $E' = \varnothing$ in the proof of Lemma 4. ▮

For a finite set E of equations we will write E also for a corresponding conjunction of equations and let the context determine whether a set or a formula is meant.

LEMMA 6. *Let t and s be ground terms and E' and E ground sets of equations such that E is finite and* $\mathrm{Con}(E') \cap (\mathrm{Con}(E) \cup \mathrm{Con}(s)) = \varnothing$. *Then*

$$\mathscr{T}_{/E' \cup E} \models (E \Rightarrow t \approx s) \Rightarrow \models (E \Rightarrow t \approx s).$$

Proof. Let E, E', s, and t be given. From $\mathscr{T}_{/E' \cup E} \models (E \Rightarrow t \approx s)$ it follows immediately that $\mathscr{T}_{/E' \cup E} \models t \approx s$ and, thus, $E' \cup E \models t \approx s$. Hence, $E \models t \approx s$ by Lemma 4; i.e., $\models (E \Rightarrow t \approx s)$. ▮

We will use the following definitions. Let φ be a quantifier-free formula and let m be a positive integer. A set of m ground substitutions Θ is an *m-corroborator* for φ if

$$\models \bigvee_{\theta \in \Theta} \varphi\theta.$$

When $\Theta = \{\theta\}$ we say that θ is a *corroborator* for φ or *corroborates* φ. The *m-corroboration* problem is the problem of determining whether a given quantifier-free formula has an *m-corroborator*.

For $x \in FV(\varphi)$, a guard for x in φ, if it exists, is a clause

$$E \Rightarrow t \approx s$$

in φ such that E and s are ground and x occurs in t. We say that

$$\bigwedge_{x \in FV(\varphi)} \psi_x$$

is a *guard* of φ if each ψ_x is a guard for x in φ; φ is called *guarded* if it has a guard.

Intuitively, in the light of Lemma 5, the notion of a Horn formula being guarded is a sufficient condition to guarantee that if there is a corroborator θ for φ then $\Sigma(\varphi\theta) = \Sigma(\varphi)$.

SREU is, by definition, the one-corroboration problem for Horn formulas. However, we only need to consider guarded Horn formulas. To see that consider a Horn formula φ, let Σ be its signature and let c be a constant in Σ. For each variable x in φ, let $\mathrm{Gr}_\Sigma(x)$ denote the Horn clause:

$$\{c' \approx c \mid c' \text{ is a constant in } \Sigma \backslash \{c\}\}$$
$$\cup \{f(c, ..., c) \approx c \mid f \text{ is a function symbol in } \Sigma\} \Rightarrow x \approx c.$$

This is a very simple but useful construction that was first used by Degtyarev and Voronkov to enforce certain solutions to be within a given signature. It is easy to see that for all terms t

$$\models \mathrm{Gr}_\Sigma(t) \Leftrightarrow t \in \mathscr{T}_\Sigma.$$

Let now ψ be the guarded Horn formula

$$\left(\bigwedge_{x \in FV(\varphi)} \mathrm{Gr}_{\Sigma}(x) \right) \wedge \varphi.$$

From Herbrand's theorem it follows that one only needs to consider corroborators in the language of φ; therefore, ψ has a corroborator if and only if φ has one.

EXAMPLE 7. A simple example of a guarded Horn formula is

$$\begin{aligned} \psi = &(E_1 \Rightarrow x \approx c_1) \\ &\wedge (E_2 \Rightarrow y \approx c_2) \\ &\wedge (\Pi_1 \Rightarrow x \approx y) \\ &\wedge (\Pi_2 \Rightarrow x \approx t \cdot y), \end{aligned}$$

where E_1, E_2, Π_1, Π_2, and t are ground terms; c_1, c_2 are constants; and "$\cdot$" is a binary function symbol written in infix notation. A guard of ψ is

$$(E_1 \Rightarrow x \approx c_1) \wedge (E_2 \Rightarrow y \approx c_2).$$

An example of a Horn formula with a common guard for all variables is

$$\begin{aligned} \varphi = &(E \Rightarrow x \cdot y \approx c) \\ &\wedge (\Pi_1 \Rightarrow x \approx y) \\ &\wedge (\Pi_2 \Rightarrow x \approx t \cdot y), \end{aligned}$$

where E, Π_1, Π_2, and t are ground and c is a constant. The guard of φ is

$$E \Rightarrow x \cdot y \approx c.$$

These formulas are of particular interest for us; see Section 6. ∎

We say that a corroborator of a disjunction φ is *partisan* if it corroborates some disjunct of φ. The main result of this section is the following theorem.

THEOREM 8 (Partisan Corroboration). *Every corroborator of a disjunction of constant-disjoint guarded Horn formulas is partisan.*

Proof. Let $\varphi = \bigvee_{i \in I} \varphi_i$, where all the φ_i's are constant-disjoint guarded Horn formulas. Let θ be a corroborator for φ. We must prove that θ corroborates φ_i for some $i \in I$.

We can assume (without loss of generality) that there exist positive integers m and n such that each φ_i has the form

$$\varphi_i = \underbrace{\bigwedge_{1 \leqslant k \leqslant m} (E_i^k \Rightarrow s_i^k \approx t_i^k)}_{\psi_i} \wedge \bigwedge_{1 \leqslant k \leqslant n} (D_i^k \Rightarrow u_i^k \approx v_i^k),$$

where ψ_i is a guard of φ_i; i.e., each E_i^k and s_i^k is ground and $FV(\varphi_i) = FV(\psi_i)$ for all $i \in I$. Let $C_i = \mathrm{Con}(\varphi_i)$ for $i \in I$. We have that

$$C_i \cap C_j = \varnothing \qquad (\forall i, j \in I, i \neq j). \tag{1}$$

Let $\Sigma = \Sigma(\varphi)$. For $i \in I$ let $\mathscr{K}_i$ denote the class of all Σ-structures that satisfy $\varphi_i\theta$, i.e,

$$\mathscr{K}_i = \{\Sigma\text{-structure } \mathscr{A} \mid \mathscr{A} \models \varphi_i\theta\}.$$

From the validity of $\varphi\theta$ it follows that each Σ-structure belongs to some $\mathscr{K}_i$.

Let now J be any subset of I such that

$$\models \psi_i\theta \qquad (\forall i \in J). \tag{2}$$

So

$$\mathrm{Con}(\varphi_i\theta) = C_i \qquad (\forall i \in J). \tag{3}$$

To see that, suppose (by contradiction) that $\mathrm{Con}(\varphi_i\theta)$ contains some $c \notin C_i$. Clearly, c belongs to some $x\theta$, where x occurs in the guard ψ_i. By Lemma 5, every constant in $x\theta$ belongs to C_i. This gives the desired contradiction.

If $I = J$ then the theorem follows by (1), (3) and Lemma 3. Assume that $I \neq J$. Below we prove the statement:

$$\text{if } \not\models \varphi_i\theta \text{ for all } i \in J \text{ then } \models \psi_i\theta \text{ for some } i \in I \backslash J. \tag{4}$$

Let now J be the *maximal* subset of I such that (2) holds. In other words, for all $i \in I \backslash J$, $\not\models \psi_i\theta$. By the contrapositive of (4) we conclude that for some $i \in J$, $\models \varphi_i\theta$. The theorem follows.

Proof of (4). Assume $\not\models \varphi_i\theta$ for all $i \in J$. Form an equation set D as follows. There is for each $i \in J$ a clause in $\varphi_i\theta$ that is not valid, and by (2) this clause is not in $\psi_i\theta$. In other words, unless J is empty, there is a mapping $f: J \to \{1, 2, ..., n\}$ such that

$$\not\models (D_i^{f(i)} \Rightarrow u_i^{f(i)} \approx v_i^{f(i)})\,\theta \qquad (\forall i \in J). \tag{5}$$

Let f be fixed and let

$$D = \bigcup_{i \in J} D_i^{f(i)}\theta.$$

(Note that $D = \varnothing$ if $J = \varnothing$.) For each mapping $g: I \backslash J \to \{1, 2, ..., m\}$ let E_g denote the set of equations

$$E_g = \bigcup_{i \in I \backslash J} E_i^{g(i)},$$

and let

$$\mathscr{A}_g = \mathscr{T}_{/E_g \cup D}.$$

We will now prove the statement:

$$\text{Fix } g\colon I\backslash J \to \{1, 2, ..., m\}. \text{ There exists } i \in I\backslash J \text{ such that } \mathscr{A}_g \in \mathscr{K}_i. \tag{6}$$

Proof. Suppose, by contradiction, that (6) does not hold. (Assume also that $J \neq \varnothing$ or else (6) holds trivially.) Then $\mathscr{A}_g \in \mathscr{K}_j$ for some $j \in J$. Fix such an appropriate j.

So $\mathscr{A}_g$ satisfies each clause in $\varphi_j\theta$ and, in particular, the following holds, call it (†):

$$\mathscr{A}_g \models (D_j^{f(j)} \Rightarrow u_j^{f(j)} \approx v_j^{f(j)})\,\theta.$$

Let $D' = D_j^{f(j)}\theta$, $D'' = \bigcup_{i \in J,\, i \neq j} D_i^{f(i)}\theta$, $u' = u_j^{f(j)}\theta$, and $v' = v_j^{f(j)}\theta$. By (3) it follows that

$$\mathrm{Con}(D', u', v') \subseteq C_j$$

and

$$\begin{aligned}\mathrm{Con}(E_g, D'') &= \mathrm{Con}(E_g) \cup \bigcup_{i \in J,\, i \neq j} \mathrm{Con}(D_i^{f(i)}\theta)\\ &\subseteq \bigcup_{i \in I\backslash J} C_i \cup \bigcup_{i \in J,\, i \neq j} C_i\\ &= \bigcup_{i \in I,\, i \neq j} C_i.\end{aligned}$$

So, by (1),

$$\mathrm{Con}(D', u', v') \cap \mathrm{Con}(E_g, D'') = \varnothing.$$

It follows, from Lemma 6 and (†), that

$$\models (D_j^{f(j)} \Rightarrow u_j^{f(j)} \approx v_j^{f(j)})\,\theta.$$

But this contradicts (5). ▮

By using (6) we can now complete the proof of (4). Suppose, by contradiction, that there is no $i \in I\backslash J$ such that $\models \psi_i\theta$. Then there is for each $i \in I\backslash J$ a clause in $\psi_i\theta$ that is not valid; i.e., there is a mapping $g\colon I\backslash J \to \{1, 2, ..., m\}$ such that

$$\not\models E_i^{g(i)} \Rightarrow s_i^{g(i)} \approx (t_i^{g(i)}\theta) \qquad (\forall i \in I\backslash J).$$

(Note that only the t_i's can be nonground.) Fix such an appropriate g.

By using (6) we know that $\mathscr{A}_g \in \mathscr{K}_i$ for some $i \in I\backslash J$. Choose such an i. So $\mathscr{A}_g$ satisfies each clause in $\varphi_i\theta$, and in particular the following holds, call it (‡):

$$\mathscr{A}_g \models E_i^{g(i)} \Rightarrow s_i^{g(i)} \approx (t_i^{g(i)}\theta).$$

But, by (3),

$$\mathrm{Con}(E_i^{g(i)}, s_i^{g(i)}) \subseteq C_i$$

and

$$\mathrm{Con}\left(\bigcup_{j \in I \setminus J,\, j \neq i} E_j^{g(j)}\right) \cup \mathrm{Con}(D) \subseteq \mathrm{Con}\left(\bigcup_{j \in I \setminus J,\, j \neq i} C_j\right) \cup \mathrm{Con}\left(\bigcup_{j \in J} C_j\right)$$
$$= \mathrm{Con}\left(\bigcup_{j \in I,\, j \neq i} C_j\right),$$

and thus, by (1),

$$\mathrm{Con}(E_i^{g(i)}, s_i^{g(i)}) \cap \mathrm{Con}\left(\bigcup_{j \in I \setminus J,\, j \neq i} E_j^{g(j)}, D\right) = \varnothing.$$

Hence, by Lemma 6 and (‡),

$$\models E_i^{g(i)} \Rightarrow s_i^{g(i)} \approx (t_i^{g(i)}\theta),$$

which contradicts our choice of g. ∎

Remark. Theorem 8, as well as its proof, remain correct if the disjunction is infinite. We will not use this generalization.

The following example illustrates why the conditions of being constant-disjoint and guarded are important and cannot, in general, be discarded. In each case there is a counterexample to the theorem.

EXAMPLE 9. Let us first consider an example where the disjuncts are guarded but not constant-disjoint. Let $\varphi(x)$ be the guarded Horn formula

$$(c \approx 0 \Rightarrow x \approx 1) \wedge (c \approx 1 \Rightarrow x \approx 0),$$

where c, 0, and 1 are constants, and let $\varphi_1 = \varphi(x_1)$, $\varphi_0 = \varphi(x_0)$, and $\psi = \varphi_1 \vee \varphi_0$, where x_1 and x_0 are distinct variables. Consider now any ground substitution θ such that $\theta(x_1) = 1$ and $\theta(x_0) = 0$. It is easy to show by case analysis that θ corroborates ψ, i.e., that

$$\models ((c \approx 0 \Rightarrow 1 \approx 1) \wedge (c \approx 1 \Rightarrow 1 \approx 0))$$
$$\vee\ ((c \approx 0 \Rightarrow 0 \approx 1) \wedge (c \approx 1 \Rightarrow 0 \approx 0)).$$

However, θ corroborates neither φ_1 nor φ_0.

Let us now consider the case when constant-disjointness is not violated but the disjuncts are not guarded. Let $\varphi_1(y, x_1, y_1)$ be the formula

$$((y \approx 0 \Rightarrow x_1 \approx y_1) \wedge (y \approx y_1 \Rightarrow x_1 \approx 0))$$

and let $\varphi_0(x_0, y_0)$ be the formula

$$((c \approx y_0 \Rightarrow x_0 \approx 1) \wedge (c \approx 1 \Rightarrow x_0 \approx y_0))$$

where c, 0, and 1 are constants and x_1, x_0, y_1, y_0, y are distinct variables. Let $\psi = \varphi_1 \vee \varphi_0$. Let θ be a ground substitution such that $\theta(x) = 1$, $\theta(x_0) = 0$, $\theta(y) = c$, $\theta(y_1) = 1$, and $\theta(y_0) = 0$. Then $\models \psi\theta$ but $\not\models \varphi_1\theta$ and $\not\models \varphi_0\theta$ (the situation is exactly the same as in the previous case). ▮

4. FROM CORROBORATION TO *M*-CORROBORATION

As Degtyarev and Voronkov (1995) have shown, the corroboration problem is undecidable. Shortly after, Voda and Komara (1995) have shown that m-corroboration is undecidable for all multiplicities m. We show that the latter result follows easily from the former result by using the Partisan Corroboration Theorem.

THEOREM 10 (Degtyarev and Voronkov). *Corroboration of guarded Horn formulas is undecidable.*

For technical reasons it will be convenient to use the following definitions. Given a set V of variables, by a (Σ, V)-expression we mean a Σ-expression all of whose variables are in V. We write $\Sigma^{(n)}$ for the constant-disjoint copy of Σ, where each constant c has been replaced by a unique constant $c^{(n)}$ not in Σ. Similarly, $V^{(n)}$ is a disjoint copy of V, where each variable x has been replaced by a unique variable $x^{(n)}$ not in V. It is also assumed that $c^{(m)} \neq c^{(n)}$ and $x^{(m)} \neq x^{(n)}$ for $m \neq n$.

We define by induction on any (Σ, V)-expression X the $(\Sigma^{(n)}, V^{(n)})$-expression $X^{(n)}$ as the one obtained from X by replacing in it each variable x with $x^{(n)}$ and each constant c with $c^{(n)}$. For any substitution θ mapping variables in V to ground Σ-terms we let $\theta^{(n)}$ denote the substitution that takes the variable $x^{(n)}$ to the term $(x\theta)^{(n)}$ for $x \in V$. So, for any (Σ, V)-expression X and natural number n,

$$(X\theta)^{(n)} = X^{(n)}\theta^{(n)}.$$

The following property holds trivially. For any (Σ, V)-sentence φ and natural number n,

$$\models \varphi \Leftrightarrow \models \varphi^{(n)}.$$

THEOREM 11. *Let φ be a guarded Horn (Σ, V)-formula and n a positive integer. Then φ has a corroborator if and only if $\bigwedge_{i=1}^{n} \varphi^{(i)}$ has an n-corroborator.*

Proof. The "⇒" direction is immediate. We prove the "⇐" direction as follows: Let $I = \{1, 2, ..., n\}$ and let ψ be the formula $\bigwedge_{i \in I} \varphi^{(i)}$. Assume that ψ has an n-corroborator $\{\theta_i \mid i \in I\}$. So

$$\models \bigvee_{i \in I} (\varphi^{(1)}\theta_i \wedge \cdots \wedge \varphi^{(i)}\theta_i \wedge \cdots \wedge \varphi^{(n)}\theta_i).$$

By using the distributive laws we can construct an equivalent formula in conjunctive normal form, including as one of the conjuncts the formula $\bigvee_{i \in I} \varphi^{(i)}\theta_i$. Hence,

$$\models \bigvee_{i \in I} \varphi^{(i)}\theta_i.$$

Let $V_i = FV(\varphi^{(i)})$ for $i \in I$. Since all the V_i's are pairwise disjoint we can let θ' be a substitution such that $\theta' \restriction V_i = \theta_i \restriction V_i$ for $i \in I$, and it follows that

$$\models \bigvee_{i \in I} \varphi^{(i)}\theta'.$$

From the Partisan Corroboration Theorem 8 follows now that $\models \varphi^{(i)}\theta'$ for some $i \in I$. Fix such an appropriate i. But then, by the fact that $\varphi^{(i)}$ is guarded and using Lemma 5, it follows that the range of $\theta' \restriction V_i$ is $\mathscr{T}_{\Sigma(\varphi^{(i)})}$, and thus, there is a substitution θ such that $\theta^{(i)} \restriction V_i = \theta' \restriction V_i$. Hence $\models \varphi^{(i)}\theta^{(i)}$ and so $\models \varphi\theta$. ∎

THEOREM 12 (Voda and Komara). *For all $n \geqslant 1$, n-corroboration is undecidable.*

Proof. Given n and φ, the construction of ψ in Theorem 11 is trivially effective. So, if we had a decision procedure (for some n) for deciding the existence of n-corroborators, we could use it to decide the existence of corroborators, but this would contradict Theorem 10. ∎

Assume that we are using an automated theorem-proving method that is based on the Herbrand theorem. Roughly, this involves a search for terms, for a given multiplicity m. The Voda–Komara theorem tells us that there is no m for which we could effectively decide when to stop our search for such terms in case they do not exist.

By using the fact that SREU is undecidable with ground left-hand sides (Plaisted, 1995) (i.e., variables occur only in positive literals in the corresponding Horn formulas) and already in the guarded case with two variables (Veanes, 1996), we can sharpen the Voda–Komara theorem as follows.

COROLLARY 13. *For all $n \geqslant 1$, n-corroboration is undecidable for guarded Horn formulas with $2n$ variables and ground negative literals.*

By a *monadic* signature or language we mean a signature or language where all function symbols have arity at most one. By *monadic* SREU or corroboration we understand the restriction of that decision problem to monadic languages. The decidability of monadic SREU is currently one of the difficult open problems related to SREU (Gurevich and Voronkov, 1997). An effectively equivalent problem is the decidability of the prenex fragment of intuitionistic logic with equality in monadic languages (Degtyarev and Voronkov 1996a). Some evidence speaks in favor of that the problem is decidable (e.g., many subcases are decidable; see Section 8). From Theorem 11 follows that

COROLLARY 14. *If monadic corroboration is undecidable, then so is monadic n-corroboration for any $n > 1$, or equivalently, if monadic n-corroboration is decidable for some $n > 1$ then so is monadic corroboration.*

5. SHIFTED PAIRING

Shifted pairing is a general technique for proving undecidability results. The term shifted pairing was introduced by Plaisted (1995). A variant of shifted pairing was used already by Hopcroft and Ullman (1979) in establishing the undecidability of the problem of testing nonemptiness of the intersection of two context-free languages. Goldfarb's (1981) proof of the undecidability of second-order unification uses also similar ideas. Finite tree automata provide a suitable abstraction level for our purposes, for formalizing this technique as a decision problem of finite tree automata.

The main result of this section is the Shifted Pairing theorem. In this section we use a binary function symbol "$\cdot$," and we write it for better readability using infix notation and assume that it associates to the right. For example, if t_1, t_2, and t_3 are terms, then the term $\cdot(t_1, \cdot(t_2, t_3))$ is written unambiguously as $t_1 \cdot t_2 \cdot t_3$.

THEOREM 15 (Shifted Pairing). *One can effectively construct two constant-disjoint tree automata*

$$A_{\mathrm{mv}} = (Q_{\mathrm{mv}}, \Sigma_{\mathrm{mv}}, R_{\mathrm{mv}}, \{q_{\mathrm{mv}}\}), \qquad A_{\mathrm{id}} = (Q_{\mathrm{id}}, \Sigma_{\mathrm{id}}, R_{\mathrm{id}}, \{q_{\mathrm{id}}\}),$$

and two ground and canonical rewrite systems

$$\Pi_1 \subseteq \mathscr{T}_{\Sigma_{\mathrm{mv}}} \times \mathscr{T}_{\Sigma_{\mathrm{id}}}, \qquad \Pi_2 \subseteq \mathscr{T}_{\Sigma_{\mathrm{mv}}} \times \mathscr{T}_{\Sigma_{\mathrm{id}}},$$

such that it is undecidable whether, given $t_0 \in \mathscr{T}_{\Sigma_{\mathrm{id}}}$, *there exists* $s \in \mathscr{L}(A_{\mathrm{mv}})$ *and* $t \in \mathscr{L}(A_{\mathrm{id}})$, *such that* $s \xrightarrow{*}_{\Pi_1} t$ *and* $t_0 \cdot s \xrightarrow{*}_{\Pi_2} t$.

The rest of this section is devoted to the proof of this theorem. We start by proving some lemmas. The proof of the theorem itself is given in Section 5.3.2.

We consider a fixed deterministic Turing machine M with *initial state* q_0, *final state* q_{f}, a *blank* symbol $\sqcup$. By $\Sigma(M)$ we denote the union of the states and tape symbols of M, including the blank symbol. All characters in $\Sigma(M)$ are considered to be *constants*. Moreover, M is only allowed to write a blank when it erases the *last* nonblank symbol on the tape. This means that IDs do not include blanks. However, overwriting the last nonblank symbol on the tape by a blank, means erasing of the last input symbol on the tape. For such a TM M we can assume, without loss of generality, that when M enters the final state then its tape is empty. Given an ID v, we let v^+ denote the string

$$v^+ = \begin{cases} \text{successor of } v, & \text{if } v \text{ is nonfinal;} \\ \varepsilon, & \text{otherwise.} \end{cases}$$

Note that the final ID of M is the unique one character string q_{f} and $q_{\mathrm{f}}^+ = \varepsilon$.

5.1. Words and Trains

We use certain nonmonadic terms to represent strings, we call such terms words. Similarly, we use certain terms that we call trains to represent sequences of strings. Let c and d be constants:

- A term s is called a *c-word* if either $s = c$, or $s = c_1 \cdot s'$ for some constant c_1 and c-word s'. The *empty c-word* is simply the constant c.
- A term t is called a *d-train of c-words* if either $t = d$, or $t = s \cdot t'$ for some c-word s and d-train t'. The *empty d-train* is simply the constant d.

We adopt convenient notation for words and trains. A c-word

$$c_1 \cdot c_2 \cdot \cdots \cdot c_n \cdot c$$

is written simply as

$$c_1 c_2 \cdots c_n \cdot c$$

and is said to *represent* the string $c_1 c_2 \cdots c_n$. When we say that a *c-word is in a set V of strings*, we mean that the string represented by that c-word is in V. Similarly, a d-train

$$(v_1 \cdot c) \cdot (v_2 \cdot c) \cdot \cdots \cdot (v_n \cdot c) \cdot d$$

is said to *represent* the string sequence

$$(v_1, v_2, ..., v_n).$$

By representing strings by words as above, one can, of course, easily represent arbitrary regular sets of strings by corresponding regular forests of words. We use this fact in the Train Lemma that is our key tool in constructing the two tree automata A_{mv} and A_{id}.

LEMMA 16 (Train Lemma). *Let V be a regular set of strings over a signature Σ of constants. Let c and d be distinct constants not in Σ. Then the set of all d-trains of c-words in V is recognized by a DTA with one final state.*

Proof. To begin with let $A_1 = (Q_1, \Sigma_1, R_1, F_1)$, where $\Sigma_1 = \Sigma \cup \{\cdot, c\}$, be a DTA that recognizes the set of all c-words in V. Next, let p be a new state, $\Sigma_2 = \Sigma_1 \cup \{d\}$, and

$$A = (Q_1 \cup \{p\}, \Sigma_2, R, \{p\}),$$

where

$$R = R_1 \cup \{d \to p\} \cup \{q \cdot p \to p \mid q \in F_1\}.$$

We prove that A is a DTA satisfying the claim. Clearly, it *is* a DTA. First, we prove the equivalence of statements 1 and 2:

1. $t \in \mathscr{L}(A)$ ($t \in \mathscr{T}_{\Sigma_2}$ and $t \xrightarrow{*}_R p$)

2. $t \in \mathscr{T}_{\Sigma_2}$ and there exist $n \geqslant 0$ and states $q_1, q_2, ..., q_n \in F_1$ such that

$$t \xrightarrow{*}_{R_1} q_1 \cdot q_2 \cdot \cdots \cdot q_n \cdot d$$
$$\rightarrow_{\{d \to p\}} q_1 \cdot q_2 \cdot \cdots \cdot q_n \cdot p$$
$$\xrightarrow{*}_{\{q \cdot p \to p \mid q \in F_1\}} p.$$

The direction from statement 2 to statement 1 is immediate. To prove the converse direction consider a reduction of t in R to p. A classical permutation argument on reductions, using the fact that $p \notin Q_1$ and $d \notin \Sigma_1$, shows that there exists a reduction, where all the rules from R_1 appear first:

$$t \xrightarrow{*}_{R_1} t' \xrightarrow{*}_{R \setminus R_1} p.$$

It follows now, by induction on the number of rewrite steps, that any reduction of t' in $R \setminus R_1$ to p must be of the desired form, proving statement 2.

Since no rule in R can introduce a "$\cdot$," the first part of the reduction in statement 2 is equivalent to saying that there exist terms $s_1, ..., s_n \in \mathscr{T}_{\Sigma_2}$ such that $t = s_1 \cdot \cdots \cdot s_n \cdot d$ and $s_i \xrightarrow{*}_{R_1} q_i$. From $s_i \xrightarrow{*}_{R_1} q_i$ it follows that d cannot occur in s_i; i.e., $s_1, ..., s_n \in \mathscr{T}_{\Sigma_1}$, and thus, $s_1, ..., s_n \in \mathscr{L}(A_1)$. Consequently, statement 2 is tantamount to saying that t is a d-train of c-words in V, and the claim follows. ∎

Let c_{id} and d_{id} be two fixed distinct constants not in $\Sigma(M)$. A *train of IDs* is a d_{id}-train of c_{id}-words representing IDs of M.

LEMMA 17. *There is a DTA* $A_{\mathrm{id}} = (Q_{\mathrm{id}}, \Sigma_{\mathrm{id}}, R_{\mathrm{id}}, \{q_{\mathrm{id}}\})$ *that recognizes the set of all trains of IDs, where* $\Sigma_{\mathrm{id}} = \Sigma(M) \cup \{\cdot, c_{\mathrm{id}}, d_{\mathrm{id}}\}$.

Proof. The set of all IDs of M is regular. Use Lemma 16. ∎

5.2. Trains of Moves

We now want to represent moves of M in such a way that we can obtain a statement corresponding to Lemma 17 for moves. A naive encoding of a move (v, v^+) as a term $(v \cdot c) \cdot (v^+ \cdot c)$ does, of course, not work for several reasons; to mention one; such terms are not recognizable.

Instead, we exploit the following information. Let (v, v^+) be a move, m the length of v, and n the length of v^+. We know that either $n = m$, $n = m + 1$ (M adds a new symbol at the end of the tape contents), or $n = m - 1$ (M erases the last nonblank symbol on the tape). We encode moves by strings of new characters, where the ith character encodes the ith characters in the components of the move. We now proceed with the formal definition.

Two new constants, denoted by $\langle a, b \rangle$ and $\langle a, b \rangle'$, respectively, are introduced for every pair of constants a and b in $\Sigma(M)$. All these new constants are assumed to be pairwise distinct. Let v be any ID of M and v^+ its successor, say

$$v = a_1 a_2 \cdots a_m,$$
$$v^+ = b_1 b_2 \cdots b_n.$$

We define $\langle v, v^+ \rangle$ as the string

$$\langle v, v^+ \rangle = \begin{cases} \langle a_1, b_1 \rangle \langle a_2, b_2 \rangle \cdots \langle a_{n-1}, b_{n-1} \rangle \langle \sqcup, b_n \rangle', & \text{if } m = n-1; \\ \langle a_1, b_1 \rangle \langle a_2, b_2 \rangle \cdots \langle a_{m-1}, b_{m-1} \rangle \langle a_m, \sqcup \rangle', & \text{if } m = n+1; \\ \langle a_1, b_1 \rangle \langle a_2, b_2 \rangle \cdots \langle a_{m-1}, b_{m-1} \rangle \langle a_m, b_m \rangle', & \text{if } m = n. \end{cases}$$

We call such a string a *move*, also.

Let c_{mv} and d_{mv} be fixed distinct new constants. A *train of moves* is a d_{mv}-train of c_{mv}-words that represent moves.

LEMMA 18. *There is a DTA* $A_{\text{mv}} = (Q_{\text{mv}}, \Sigma_{\text{mv}}, R_{\text{mv}}, \{q_{\text{mv}}\})$ *that recognizes the set of all trains of moves, where*

$$\Sigma_{\text{mv}} = \{\langle a, b \rangle, \langle a, b \rangle' \mid a, b \in \Sigma(M)\} \cup \{\cdot, c_{\text{mv}}, d_{\text{mv}}\}.$$

Proof. Follows from the train lemma 16 and the fact that the set of moves is regular. The important property that is exploited here is that only a fixed size substring of an ID is changed by a move.

For example, the set of all moves corresponding to computation steps that do not change the last tape symbol can be described by the regular set of strings

$$V^* V_\delta V^* V',$$

where V_δ is a certain finite set of three-character or two-character strings, constructed from the transition function of M; e.g., if M, upon reading the symbol a in state q, writes the symbol a', moves right, and enters state q', then $\langle q, a' \rangle \langle a, q' \rangle$ is in V_δ. The set V consists all constants $\langle a, a \rangle$ such that a is an input symbol of M, and V' is the set of all constants $\langle a, a \rangle'$ such that a is an input symbol of M. The other cases are similar. ▮

5.3. Main Construction

Given a nonempty train t of moves, say

$$t = (\langle v_1, v_1^+ \rangle \cdot c_{\text{mv}}) \cdot (\langle v_2, v_2^+ \rangle \cdot c_{\text{mv}}) \cdot \cdots \cdot (\langle v_{k-1} v_{k-1}^+ \rangle \cdot c_{\text{mv}}) \cdot (\langle v_k, v_k^+ \rangle \cdot c_{\text{mv}}) \cdot d_{\text{mv}},$$

define the *first projection* of t as the train of IDs

$$\pi_1(t) = (v_1 \cdot c_{\text{id}}) \cdot (v_2 \cdot c_{\text{id}}) \cdot \cdots \cdot (v_{k-1} \cdot c_{\text{id}}) \cdot (v_k \cdot c_{\text{id}}) \cdot d_{\text{id}}$$

and the *second projection* of t as the train

$$\pi_2(t) = \begin{cases} (v_1^+ \cdot c_{\text{id}}) \cdot (v_2^+ \cdot c_{\text{id}}) \cdot \cdots \cdot (v_{k-1}^+ \cdot c_{\text{id}}) \cdot d_{\text{id}}, & \text{if } v_k = q_{\text{f}}; \\ (v_1^+ \cdot c_{\text{id}}) \cdot (v_2^+ \cdot c_{\text{id}}) \cdot \cdots \cdot (v_{k-1}^+ \cdot c_{\text{id}}) \cdot (v_k^+ \cdot c_{\text{id}}) \cdot d_{\text{id}}, & \text{otherwise.} \end{cases}$$

We say that t is a *shifted pairing train* if t is a train of moves such that

$$\pi_1(t) = (v_1 \cdot c_{\mathrm{id}}) \cdot \pi_2(t)$$

and we refer to v_1 as the *first ID* of t. Recall that q_0 is the initial state of M.

LEMMA 19. *Let v_0 be an input string for M. Then there exists a shifted pairing train t with first ID $q_0 v_0$ if and only if M accepts v_0.*

Proof. Let v_0 be given and t a train of moves as above, with $v_1 = q_0 v_0$. If t is a shifted pairing train then the second projection must be shorter than the first one, and thus $v_k = q_{\mathrm{f}}$ and

$$(v_1, v_2, v_3, ..., v_{k-1}, v_k) = (q_0 v_0, v_1^+, v_2^+, ..., v_{k-2}^+, v_{k-1}^+)$$

which is tantamount to saying that the first projection of t represents a valid computation of M with input v_0; i.e., M accepts v_0. The proof of the converse direction is similar. ▮

5.3.1. *The rewrite systems Π_1 and Π_2.* The system Π_1 contains all the following rules:

1. For all $a, b \in \Sigma(M)$, the rule $\langle a, b\rangle \to a$.
2. For all $a, b \in \Sigma(M)$ such that $a \neq \sqcup$, the rule $\langle a, b\rangle' \cdot v_{\mathrm{mv}} \to a \cdot c_{\mathrm{id}}$.
3. For all $b \in \Sigma(M)$, the rule $\langle \sqcup, b\rangle' \cdot c_{\mathrm{mv}} \to c_{\mathrm{id}}$.
4. The rule $d_{\mathrm{mv}} \to d_{\mathrm{id}}$.

LEMMA 20. *The rewrite system Π_1 is canonical and $\Pi_1 \subseteq \mathscr{T}_{\Sigma_{\mathrm{mv}}} \times \mathscr{T}_{\Sigma_{\mathrm{id}}}$.*

Proof. It is easy to check that the rules in Π_1 form a reduced set of rules and Π_1 is therefore canonical. ▮

Hence, we have the following relation between Π_1 and the notion of first projection.

LEMMA 21. *For all trains s of moves and all trains t of IDs, $s \xrightarrow{*}_{\Pi_1} t$ if and only if $t = \pi_1(s)$.*

Proof. Let s and t be given. By Lemma 20 t is irreducible in Π_1 because Σ_{mv} and Σ_{id} are constant-disjoint. So, $s \xrightarrow{*}_{\Pi_1} t$ if and only if $s\downarrow_{\Pi_1} = t$. It remains to be checked that indeed $s\downarrow_{\Pi_1} = \pi_1(s)$, which is straightforward. ▮

The system Π_2 contains all the following rules:

1. For all $a, b \in \Sigma(M)$, the rule $\langle a, b\rangle \to b$.
2. For all $a, b \in \Sigma(M)$ such that $b \neq \sqcup$, the rule $\langle a, b\rangle' \cdot c_{\mathrm{mv}} \to b \cdot c_{\mathrm{id}}$.
3. For all $a \in \Sigma(M)$ such that $a \neq q_{\mathrm{f}}$, the rule $\langle a, \sqcup\rangle' \cdot c_{\mathrm{mv}} \to c_{\mathrm{id}}$.
4. The rule $(\langle q_{\mathrm{f}}, \sqcup\rangle' \cdot c_{\mathrm{mv}}) \cdot d_{\mathrm{mv}} \to d_{\mathrm{id}}$.

Again, one can easily check that the rules in 1–4 form a reduced rule set.

LEMMA 22. *The rewrite system Π_2 is canonical and $\Pi_2 \subseteq \mathscr{T}_{\Sigma_{\mathrm{mv}}} \times \mathscr{T}_{\Sigma_{\mathrm{id}}}$.*

LEMMA 23. *For all trains s of moves and all IDs v, $(v \cdot c_{\mathrm{id}}) \cdot s \xrightarrow{*}_{\Pi_2} \pi_1(s)$ if and only if s is a shifted pairing train with first ID v.*

Proof. Let s and v be given and assume that $(v \cdot c_{\mathrm{id}}) \cdot s \xrightarrow{*}_{\Pi_2} \pi_1(s)$, say

$$s = (\langle v_1, v_1^+ \rangle \cdot c_{\mathrm{mv}}) \cdot \cdots \cdot (\langle v_{k-1}, v_{k-1}^+ \rangle \cdot c_{\mathrm{mv}}) \cdot (\langle v_k, v_k^+ \rangle \cdot c_{\mathrm{mv}}) \cdot d_{\mathrm{mv}}.$$

So

$$\pi_1(s) = (v_1 \cdot c_{\mathrm{id}}) \cdot (v_2 \cdot c_{\mathrm{id}}) \cdot \cdots \cdot (v_{k-1} \cdot c_{\mathrm{id}}) \cdot (v_k \cdot c_{\mathrm{id}}) \cdot d_{\mathrm{id}}.$$

By Lemma 22, $((v \cdot c_{\mathrm{id}}) \cdot s) \downarrow_{\Pi_2} = \pi_1(s)$, and thus, $v_1 = v$ and

$$s \downarrow_{\Pi_2} = (v_2 \cdot c_{\mathrm{id}}) \cdot \cdots \cdot (v_{k-1} \cdot c_{\mathrm{id}}) \cdot (v_k \cdot c_{\mathrm{id}}) \cdot d_{\mathrm{id}}.$$

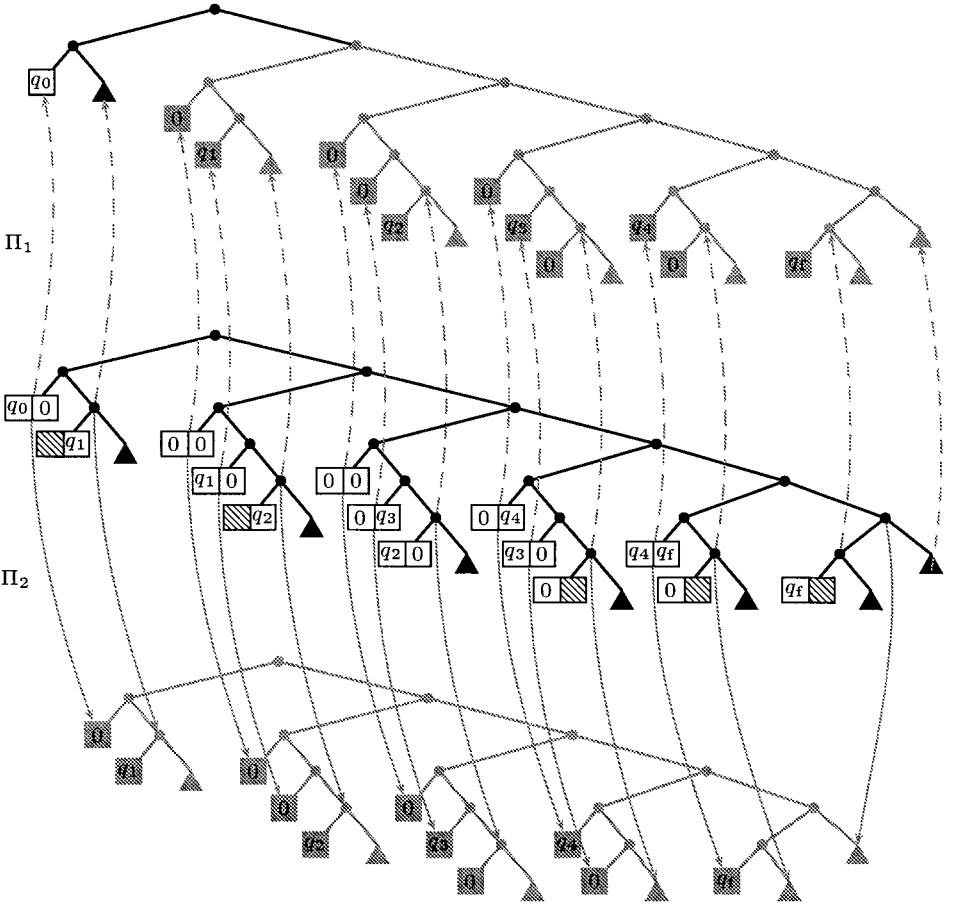

FIG. 2. Example of shifted pairing. Consider a Turing machine M that, given an empty input string, writes two 0's and then simply erases them. A valid computation of M can have the form $(q_0, 0q_1, 00q_2, 0q_30, q_40, q_f)$. The corresponding train of moves is the middle tree in the figure, say s, with the tree above s as the first projection of s, and the tree below s as the second projection of s. The arrows illustrate in a precise way, how Π_1 and Π_2 reduce s to its first projection and second projection, respectively.

On the other hand, from the definition of Π_2 it follows that all the rules can be applied simultaneously which implies that

$$(\langle v_i, v_i^+ \rangle \cdot c_{\mathrm{mv}}) \downarrow_{\Pi_2} = v_{i+1} \cdot c_{\mathrm{id}} \qquad \text{for} \quad 1 \leqslant i < k, \tag{7}$$

and the last word of s must be removed:

$$((\langle v_k, v_k^+ \rangle \cdot c_{\mathrm{mv}}) \cdot d_{\mathrm{mv}}) \downarrow_{\Pi_2} = d_{\mathrm{id}}. \tag{8}$$

From (7) follows that $v_i^+ = v_{i+1}$ for $1 \leqslant i < k$ and from (8) it follows that the rule $(\langle q_{\mathrm{f}}, \sqcup \rangle' \cdot c_{\mathrm{mv}}) \cdot d_{\mathrm{mv}} \to d_{\mathrm{id}}$ is used, and thus, $v_k = q_{\mathrm{f}}$. Hence, s is a shifted pairing train with first ID v.

The proof of the converse direction is analogous. ▌

5.3.2. *Proof of the shifted Pairing Theorem.* Let M in the above construction be a universal Turing machine. Let A_{id} and A_{mv} be constant-disjoint DTAs given by Lemma 17 and Lemma 18, respectively. The claim in Theorem 15 is a consequence of the equivalence of the following statements. The additional conditions on the rewrite systems Π_1 and Π_2 follow from Lemma 20 and Lemma 22:

1. M accepts v_0.
2. There exists $s \in \mathscr{L}(A_{\mathrm{mv}})$ such that $(q_0 v_0 \cdot c_{\mathrm{id}}) \cdot s \xrightarrow{*}_{\Pi_2} \pi_1(s)$.
3. There exist $s \in \mathscr{L}(A_{\mathrm{mv}})$ and $t \in \mathscr{L}(A_{\mathrm{id}})$, such that $s \xrightarrow{*}_{\Pi_1} t$ and $(q_0 v_0 \cdot c_{\mathrm{id}}) \cdot s \xrightarrow{*}_{\Pi_2} t$.

Statements 1 and 2 are equivalent by Lemma 19 and Lemma 23. Statements 2 and 3 are equivalent by Lemma 21. ▌

See Fig. 2 for a concrete example.

6. APPLICATIONS OF PARTISAN CORROBORATION THEOREM AND SHIFTED PAIRING THEOREM

The Shifted Pairing Theorem is used here to give a very elementary undecidability proof of SREU. The latter result is then used, in combination with the Partisan Corroboration Theorem to improve upon the undecidability result of n-corroboration for arbitrary n.

6.1. *Undecidability of SREU: Minimal Case.*

Consider fixed constant-disjoint DTAs $A_{\mathrm{mv}} = (Q_{\mathrm{mv}}, \Sigma_{\mathrm{mv}}, R_{\mathrm{mv}}, \{q_{\mathrm{mv}}\})$ and $A_{\mathrm{id}} = (Q_{\mathrm{id}}, \Sigma_{\mathrm{id}}, R_{\mathrm{id}}, \{q_{\mathrm{id}}\})$, a binary function symbol $\cdot$, and ground canonical rewrite systems Π_1 and Π_2 given by the Shifted Pairing Theorem 15. Let q be a new state and A the tree automaton (Q, Σ, R, F), where

$$\begin{aligned} Q &= Q_{\mathrm{mv}} \cup Q_{\mathrm{id}} \cup \{q\}, \\ \Sigma &= \Sigma_{\mathrm{mv}} \cup \Sigma_{\mathrm{id}}, \\ R &= R_{\mathrm{mv}} \cup R_{\mathrm{id}} \cup \{q_{\mathrm{mv}} \cdot q_{\mathrm{id}} \to q\}, \\ F &= \{q\}. \end{aligned}$$

Obviously, A is still a *deterministic* tree automaton, because A_{mv} and A_{id} are constant-disjoint and deterministic. We have the following property as a direct consequence of the constant-disjointness of A_{id} and A_{mv}.

LEMMA 24. *For all ground terms s and t, $s \cdot t \xrightarrow{*}_R q$ if and only if $s \xrightarrow{*}_{R_{\mathrm{mv}}} q_{\mathrm{mv}}$ and $t \xrightarrow{*}_{R_{\mathrm{id}}} q_{\mathrm{id}}$.*

We can now prove the following result. Recall that a *rigid equation* is an expression $E \vdash^{\mathrm{r}} s \approx t$, where E is a finite set of equations and s and t are terms. A ground substitution θ **solves** $E \vdash^{\mathrm{r}} s \approx t$ if θ corroborates $E \Rightarrow s \approx t$. *SREU* is the problem of deciding if there exists a θ that solves all members in a given finite set of rigid equations.

THEOREM 25. *There is an integer n, such that SREU is undecidable under the following restrictions*:

1. (Plaisted). *The left-hand sides are ground,*
2. *the left-hand sides have at most n symbols,*
3. *there are at most two variables each occurring at most three times, and*
4. *there are at most three rigid equations.*

Proof. Let $S_{t_0}(x, y)$ be the following system of rigid equations, where t_0 is a given ground term over Σ_{id}:

$$S_{t_0}(x, y) = \begin{cases} R^{\approx} \vdash^{\mathrm{r}} x \cdot y \approx q, \\ \Pi_1^{\approx} \vdash^{\mathrm{r}} x \approx y \\ \Pi_2^{\approx} \vdash^{\mathrm{r}} t_0 \cdot x \approx y. \end{cases}$$

Let θ be a ground substitution with $x\theta = s$ and $y\theta = t$. Since all the left-hand sides are canonical rewrite systems, by using Birkhoff's theorem, we get that θ solves $S_{t_0}(x, y)$ if and only if

$$(s \cdot t)\downarrow_R = q\downarrow_R, \qquad s\downarrow_{\Pi_1} = t\downarrow_{\Pi_1}, \quad t_0 \cdot s\downarrow_{\Pi_2} = t\downarrow_{\Pi_2}.$$

By using Lemma 24 and that q is irreducible in R, this is equivalent to

$$s\downarrow_{R_{\mathrm{mv}}} = q_{\mathrm{mv}}, \qquad t\downarrow_{R_{\mathrm{id}}} = q_{\mathrm{id}}, \quad s\downarrow_{\Pi_1} = t\downarrow_{\Pi_1}, \quad t_0 \cdot s\downarrow_{\Pi_2} = t\downarrow_{\Pi_2}. \tag{9}$$

The first two facts in (9) imply that $s \in \mathscr{T}_{\Sigma_{\mathrm{mv}} \cup \mathcal{Q}_{\mathrm{mv}}}$ and $t \in \mathscr{T}_{\Sigma_{\mathrm{id}} \cup \mathcal{Q}_{\mathrm{id}}}$. In particular, s and t are constant-disjoint. At the same time, $\Pi_1^{\approx} \models s \approx t$ implies (see Lemma 5) that $\mathrm{Con}(s) \subseteq \mathrm{Con}(\Pi_1, t)$ and $\mathrm{Con}(t) \subseteq \mathrm{Con}(\Pi_1, s)$. Hence $\mathrm{Con}(s, t) \subseteq \mathrm{Con}(\Pi_1) \subseteq \Sigma_{\mathrm{id}} \cup \Sigma_{\mathrm{mv}}$. So (9) implies that $s \in \mathscr{T}_{\Sigma_{\mathrm{mv}}}$ and $t \in \mathscr{T}_{\Sigma_{id}}$, and therefore, (9) is equivalent to

$$s \in \mathscr{L}(A_{\mathrm{mv}}), \qquad t \in \mathscr{L}(A_{\mathrm{id}}), \qquad s\downarrow_{\Pi_1} = t\downarrow_{\Pi_1}, \qquad t_0 \cdot s\downarrow_{\Pi_2} = t\downarrow_{\Pi_2}. \tag{10}$$

But t is irreducible in both Π_1 and Π_2, so (10) is equivalent to

$$s \in \mathscr{L}(A_{\mathrm{mv}}), \qquad t \in \mathscr{L}(A_{\mathrm{id}}), \quad s \xrightarrow{*}_{\Pi_1} t, \quad t_0 \cdot s \xrightarrow{*}_{\Pi_2} t. \tag{11}$$

By the Shifted Pairing Theorem 15, the problem of existence of such s and t for a given t_0 is undecidable, and thus, so is the solvability of $S_{t_0}(x, y)$ for a given t_0. The additional conditions are simply properties of $S_{t_0}(x, y)$ and n can be chosen to be any integer greater than the number of symbols in the left-hand sides of the rigid equations in $S_{t_0}(x, y)$. ▮

Undecidability proofs of SREU. Degtyarev and Voronkov's (1995) original proof of the undecidability of SREU was by reduction of Baaz's (1993) monadic semi-unification problem. This proof was followed by other proofs by Degtyarev and Voronkov, first by reducing second-order unification to SREU (1996c), and then by reducing Hilbert's tenth problem to SREU (1996b). The undecidability of second-order unification was proved by Goldfarb (1981). Plaisted (1995) reduced Post's Correspondence Problem to SREU. From his proof follows that SREU is undecidable already with ground left-hand sides. Veanes (1996) improved that construction by using the halting problem for Turing machines and showed that two variables and one binary function symbol is enough to obtain undecidability. Here we have shown that, in addition, already three rigid equations suffice for the undecidability.

6.2. *Undecidability of m-Corroboration: Minimal Case*

Consider the system $S_{t_0}(x, y)$ of rigid equations in the proof of Theorem 25 and let φ_{t_0} denote the corresponding guarded Horn formula:

$$(R^{\approx} \Rightarrow x \cdot y \approx c) \wedge (\Pi_1^{\approx} \Rightarrow x \approx y) \wedge (\Pi_2^{\approx} \Rightarrow t_0 \cdot x \approx y).$$

A formula is *ground negative* if all negatively occurring atoms in it are ground. For example φ_{t_0} is ground negative.

THEOREM 26. *For all $m \geqslant 1$, m-corroboration is undecidable for ground negative guarded Horn formulas with at most $2m$ variables and at most $3m$ clauses.*

Proof. Given m and t_0, construct the formula $\psi = \bigwedge_{1 \leqslant i \leqslant m} \varphi_{t_0}^{(i)}$. By Theorem 11, ψ has an m-corroborator if and only if φ_{t_0} has a corroborator. The rest follows from Theorem 25. ▮

7. RELATIONS TO INTUITIONISTIC LOGIC

The decision problems in intuitionistic logic have not been as thoroughly studied as the corresponding problems in classical logic (Börger, Grädel, and Gurevich, 1997). In particular, new results about the *prenex fragment* of intuitionistic logic (i.e., closed prenex formulas that are intuitionistically provable) have been obtained recently by Degtyarev and Voronkov in (1996b, 1996c, 1996a) and Voronkov (1996). Some of these results are:

1. Decidability, and in particular PSPACE-completeness, of the prenex fragment of intuitionistic logic *without* equality (Degtyarev and Voronkov, 1996a).

2. Prenex fragment of intuitionistic logic *with* equality but *without* function symbols is PSPACE-complete (Degtyarev and Voronkov, 1996a). Decidability of this fragment was proved by Orevkov (1976).

3. Prenex fragment of intuitionistic logic with equality in the language with one unary function symbol is decidable (Degtyarev and Voronkov, 1996a).

4. $\exists^*$-fragment of intuitionistic logic with equality is undecidable (Degtyarev and Voronkov, 1996b, 1996c).

In some of the above results, the corresponding result has first been obtained for a fragment of SREU with similar restrictions. The undecidability of the $\exists^*$-fragment was improved by Veanes (1996) by showing that already the

5. $\exists\exists$-fragment of intuitionistic logic with equality is undecidable.

Given a rigid equation $E \vdash^{\mathrm{r}} s \approx t$, let $\mathscr{F}(E \vdash^{\mathrm{r}} s \approx t)$ denote the implication in intuitionistic logic with equality:

$$\left(\bigwedge_{e \in E} e \right) \Rightarrow s \approx t.$$

Given a system S of rigid equations, let $\mathscr{F}(S)$ denote the conjunction:

$$\bigwedge_{E \vdash^{\mathrm{r}} s \approx t \in S} \mathscr{F}(E \vdash^{\mathrm{r}} s \approx t).$$

Provability in intuitionsitic logic with equality is related to SREU through the following lemma (Degtyarev and Voronkov, 1996c).

LEMMA 27. *A system $S(\mathbf{x})$ of rigid equations is solvable if and only if the formula $\exists \mathbf{x} \mathscr{F}(S(\mathbf{x}))$ is provable in intuitionistic logic with equality.*

By using Theorem 25 and Lemma 27, we obtain the following sharpening of the result in (Veanes 1996).

COROLLARY 28. *There is an integer n such that the $\exists\exists$-fragment of intuitionistic logic with equality is undecidable already under the following restrictions*:

1. *The only logical connectives are $\wedge$ and at most three $\Rightarrow$'s.*
2. *The antecedents of all implications are ground and have less than n symbols.*

In contrast, Degtyarev, Gurevich, Narendran, Veanes, and Voronkov (1998b) have shown that the

6. $\forall^*\exists\forall^*$-fragment of intuitionistic logic with equality is decidable.

Note that statements 5 and 6 imply a complete classification of the decidability of the prenex fragment of intuitionistic logic with equality in terms of the quantifier prefix.

7.1. Other Fragments

Decidability problems for other fragments of intuitionistic logic have been studied by Orevkov (1965, 1976), Mints (1967), Statman (1979), and Lifschitz

(1967). Orevkov (1965) proves that the $\neg\neg\forall\exists$-fragment of intuitionistic logic with function symbols is undecidable. Lifschitz (1967) proves that intuitionistic logic with equality and without function symbols is undecidable, i.e., that the pure constructive theory of equality is undecidable. Orevkov (1976) shows decidability of some fragments (that are close to the prenex fragment) of intuitionistic logic with equality. Statman (1979) proves that the intuitionistic propositional logic is PSPACE-complete.

7.2. A Remark about Proof Search

Proof search in intuitionistic logic with equality is closely connected with SREU, and, unlike in the classical case, the handling of SREU is in fact *unavoidable* in that context, which is clearly illustrated by Lemma 27. Voronkov (1996) considers a particular sequent calculus-based proof system $\mathrm{LJ}^{\approx}$. A part of that system is shown in Fig. 3. A *proof skeleton* in $\mathrm{LJ}^{\approx}$ is obtained from a proof in $\mathrm{LJ}^{\approx}$ by erasing all sequents and keeping only a tree decorated with rule names. See Fig. 4.

Skeleton instantiation is the decision problem of the existence of a proof of a given formula with a given (proof) skeleton. Voronkov (1996) shows that SREU is polynomial time equivalent to skeleton instantiation in $\mathrm{LJ}^{\approx}$. So in particular, the skeleton instantiation problem in $\mathrm{LJ}^{\approx}$ is undecidable. Lemma 27 and the system of rigid equations constructed in the proof of Theorem 25 can be used to exhibit a *fixed* skeleton for which the skeleton instantiation problem is undecidable. Such a "universal" skeleton is illustrated in Fig 4.

$$\frac{\Gamma, \phi, \psi, \Delta \to \chi}{\Gamma, \phi \wedge \psi, \Delta \to \chi}\ (\wedge\to) \qquad \frac{\Gamma \to \phi \quad \Gamma \to \psi}{\Gamma \to \phi \wedge \psi}\ (\to \wedge)$$

$$\frac{\Gamma, \phi, \Delta \to \chi \quad \Gamma, \psi, \Delta \to \chi}{\Gamma, \phi \vee \psi, \Delta \to \chi}\ (\vee\to) \qquad \frac{\Gamma \to \phi}{\Gamma \to \phi \vee \psi}\ (\to \vee_1)$$

$$\frac{\Gamma, \psi, \Delta \to \chi \quad \Gamma, \phi \Rightarrow \psi, \Delta \to \phi}{\Gamma, \phi \Rightarrow \psi, \Delta \to \chi}\ (\Rightarrow\to) \qquad \frac{\Gamma \to \psi}{\Gamma \to \phi \vee \psi}\ (\to \vee_2)$$

$$\frac{\phi, \Gamma \to \psi}{\Gamma \to \phi \Rightarrow \psi}\ (\to\Rightarrow)$$

$$\frac{\Gamma, \phi\{x \mapsto t\}, \forall x\phi, \Delta \to \chi}{\Gamma, \forall x\phi, \Delta \to \chi}\ (\forall\to) \qquad \frac{\Gamma \to \phi\{x \mapsto y\}}{\Gamma \to \forall x\phi}\ (\to \forall)$$

$$\frac{\Gamma, \phi\{x \mapsto y\}, \Delta \to \chi}{\Gamma, \exists x\phi, \Delta \to \chi}\ (\exists\to) \qquad \frac{\Gamma \to \phi\{x \mapsto t\}}{\Gamma \to \exists x\phi}\ (\to \exists)$$

FIG. 3. The propositional and quantifier inference rules of $\mathrm{LJ}^{\approx}$. Here Γ and Δ are multisets of (side) formulas. In the rules $(\exists\to)$ and $(\to\forall)$ the variable y does not occur free in the conclusions of the rules.

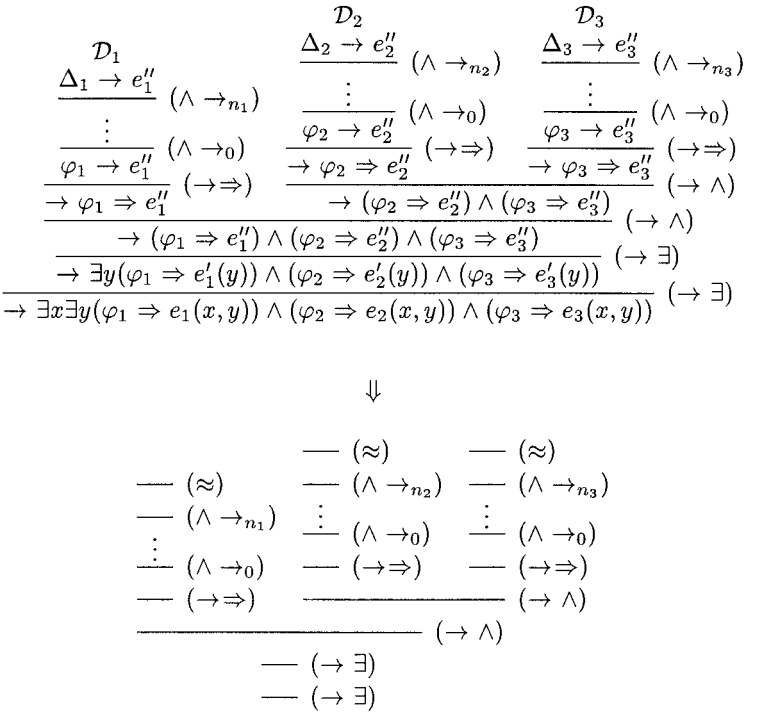

FIG. 4. The upper part of the figure shows a possible proof tree in $\mathrm{LJ}^{\approx}$ of the formula $\exists x\, \exists y\, (\varphi_1 \Rightarrow e_1(x, y)) \wedge (\varphi_2 \Rightarrow e_2(x, y)) \wedge (\varphi_3 \Rightarrow e_3(x, y))$. This formula denotes the formula $\exists x\, \exists y\, \mathscr{F}(S_{t_0}(x, y))$, where $S_{t_0}(x, y)$ is the system of rigid equations taken from the proof of Theorem 25. Here the Δ_i's are multisets of equations, the $\mathscr{D}_i$'s are subproofs consisting of inference rules involving equality only, and each n_i is the size of φ_i. The corresponding proof skeleton is shown in the lower part of the figure.

8. CURRENT STATUS OF SREU AND OPEN PROBLEMS

Here we briefly summarize the current status of SREU and mention some open problems. Many related results are already mentioned above. The first decidability proof of rigid E-unification is given by Gallier, Narendran, Plaisted, and Snyder (1988). De Kogel (1995) has presented a simpler proof, without computational complexity considerations. We start with the solved cases:

- Rigid E-unification with ground left-hand side is NP-complete (Kozen, 1981). Rigid E-unification in general is NP-complete and there exist finite complete sets of unifiers (Gallier, Narendran, Plaisted, and Snyder, 1990; Gallier *et al.*, 1988). (Here completeness has a very special meaning, differing from its meaning in the context of (nonrigid) E-unification.)

- Rigid E-unification with one variable, or more generally, SREU with one variable and a *fixed* number of rigid equations is P-complete (Degtyarev *et al.*, 1998b).

- If all function symbols have arity $\leqslant 1$ (the *monadic* case) then it follows that SREU is PSPACE-hard (Goubault, 1994). If only one unary function symbol is

allowed then the problem is decidable (Degtyarev, Matiyasevich, and Voronkov, 1996). If only constants are allowed then the problem is NP-complete (Degtyarev, Matiyasevich, and Voronkov, 1996), assuming that there are at least two constants.

- About the monadic case it is known that if there are more than one unary function symbols then SREU is decidable if and only if it is decidable with just two unary function symbols (Degtyarev, Matiyasevich, and Voronkov, 1996).

- If the left-hand sides are ground then the monadic case is decidable (Gurevich and Voronkov, 1997) and, in fact, PSPACE-complete (Cortier, Ganzinger, Jacquemard, and Veanes, 1999). A more general problem is shown to be decidable in (Ganzinger *et al.*, 1998). Monadic SREU with one variable is PSPACE-complete (Gurevich and Voronkov, 1997).

- The word equation solving (Makanin, 1977), which is an extremely hard problem, can be reduced to monadic SREU (Degtyarev, Matiyasevich, and Voronkov, 1996).

- Monadic SREU is equivalent to a nontrivial extension of word equations (Gurevich and Voronkov, 1997).

- Monadic SREU is equivalent to the decidability problem of the prenex fragment of intuitionistic logic with equality with function symbols of arity $\leqslant 1$ (Degtyarev and Voronkov, 1996a).

- In general SREU is undecidable (Degtyarev and Voronkov, 1995). Moreover, SREU is undecidable under the following restrictions:

  — The left-hand sides of the rigid equations are ground (Plaisted, 1995).

  — Furthermore, there are only two variables (Veanes 1996) and three rigid equations with ground left-hand sides of bounded size.

- SREU with one variable is decidable, in fact, EXPTIME-complete (Degtyarev *et al.* 1998b). Further decidable cases are proved in (Degtyarev, Gurevich, Narendran, Veanes, and Voronkov 1998a) and (Cortier *et al.*, 1999).

- SREU is polynomial time equivalent with second-order unification (Levy, 1998; Veanes, 1998).

The unsolved cases are:

- Decidability of monadic SREU.
- Decidability of SREU with two rigid equations.

Both problems are highly nontrivial. An intriguing problem is also the corroboration problem with a given strategy. In particular, the open problem was posed by Voronkov (1997):

Does there exist a computable strategy f with which the corroboration problem is decidable?

Further problems related to SREU and the Herbrand theorem are discussed in (Voronkov, 1998b, 1998a).

ACKNOWLEDGMENTS

We thank Andrei Voronkov and Anatoli Degtyarev for many valuable discussions. We thank Florent Jacquemard for useful comments on a preliminary version of this paper. The comments and suggestions of an anonymous referee greatly improved the final manuscript.

Received July 22, 1997; final manuscript received January 6, 1999

REFERENCES

Baaz, M. (1993), Note on the existence of most general semi-unifiers, *in* "Arithmetic, Proof Theory and Computation Complexity, Oxford Logic Guides," Vol. 23, pp. 20–29, Oxford Univ. Press, London.

Birkhoff, G. (1935), On the structure of abstract algebras, *Proc. Cambridge Phil. Soc.* **31**, 433–454.

Börger, E., Grädel, E., and Gurevich, Y. (1997), "The Classical Decision Problem," Springer-Verlag, New York/Berlin.

Chang, C., and Keisler, H. (1990), "Models Theory," 3rd ed., North-Holland, Amsterdam.

Cortier, V., Ganzinger, H., Jacquemard, F., and Veanes, M. (1999), Decidable fragments of simultaneous rigid reachability, *in* "Proc. ICALP'99."

De Kogel, E. (1995), Rigid *E*-unification simplified, *in* "Theorem Proving with Analytic Tableaux and Related Methods" (P. Baumgartner and J. Posegga, Eds.), Lecture Notes in Artificial Intelligence Vol. 918, pp. 17–30, Schloß Rheinfels, St. Goar, Germany.

Degtyarev, A., and Voronkov, A. (1995), "Simultaneous Rigid *E*-Unification Is Undecidable," UPMAIL Technical Report 105, Uppsala University Computing Science Department.

Degtyarev, A., and Voronkov, A. (1996a), Decidability problems for the prenex fragment of intuitionistic logic, *in* "Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)," pp. 503–512, IEEE Comput. Soc. Press, New Brunswick, NJ.

Degtyarev, A., and Voronkov, A. (1996b), Simultaneous rigid *E*-unification is undecidable, *in* "Computer Science Logic. 9th International Workshop, CSL'95, Paderborn, Germany, September 1995" (H. Kleine Büning, Ed.), Lecture Notes in Computer Science, Vol. 1092, pp. 178–190.

Degtyarev, A., and Voronkov, A. (1996c), The undecidability of simultaneous rigid *E*-unification, *Theor. Comput. Sci.* **166**(1–2), 291–300.

Degtyarev, A. Gurevich, Y., and Voronkov, A. (1996), Herbrand's theorem and equational reasoning: Problems and solutions, *in* "Bulletin of the European Association for Theoretical Computer Science" Vol. 60, the "Logic in Computer Science" column.

Degtyarev, A., Gurevich, Y., Narendran, P., Veanes, M., and Voronkov, A. (1998a), Decidability and complexity of simultaneous rigid *E*-unification with one variable and related results, *Theor. Comput. Sci.*, to appear.

Degtyarev, A. Gurevich, Y., Narendran, P. Veanes, M., and Voronkov, A. (1998b), The decidability of simultaneous rigid *E*-unification with one variable, *in* "Rewriting Techniques and Applications" (T. Nipkow, Ed.), Lecture Notes in Computer Science, Vol. 1379, pp. 181–195, Springer-Verlag, New York/Berlin.

Degtyarev, A. Matiyasevich, Y., and Voronkov, A. (1996), Simultaneous rigid *E*-unification and related algorithmic problems, *in* "Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS'96)," pp. 494–502, IEEE Computer Society Press, New Brunswick, NJ.

Dershowitz, N., and Jouannaud, J.-P. (1990), Rewrite systems, *in* "Handbook of Theoretical Computer Science" (J. Van Leeuwen, Ed.), Formal Methods and Semantics, Vol. B, chapter 6, pp. 243–309, North Holland, Amsterdam.

Gallier, J. Narendran, P. Plaisted, D., and Snyder, W. (1988), Rigid *E*-unification is NP-complete, *in* "Proc. IEEE Conference on Logic in Computer Science (LICS)," pp. 338–346, IEEE Computer Society Press.

Gallier, J., Narendran, P., Plaisted, D., and Snyder, W. (1990), Rigid *E*-unification: NP-completeness and applications to equational matings, *Inform. and Comput.* **87**(1/2), 129–195.

Gallier, J. Raatz, S., and Snyder, W. (1987), Theorem proving using rigid *E*-unification: Equational matings, *in* "Proc. IEEE Conference on Logic in Computer Science (LICS)," pp. 338–346, IEEE Comput. Soc., New York.

Ganzinger, H. Jacquemard, F., and Veanes, M. (1998), Rigid reachability, *in* "Advances in Computing Science-ASIAN'98, 4th Asian Computing Science Conference, Manila, The Philippines, December 1998, Proceedings" (J. Hsiang and A. Ohori, Eds.), Lecture Notes in Computer Science, Vol. 1538, pp. 4–21, Springer-Verlag.

Goldfarb, W. (1981), The undecidability of the second-order unification problem, *Theoretical Computer Science* **13**, 225–230.

Goubault, J. (1994), Rigid **E**-unifiability is DEXPTIME-complete, *in* "Proc. IEEE Conference on Logic in Computer Science (LICS)" IEEE Comput. Soc., Los Alamitos, CA.

Gurevich, Y., and Voronkov, A. (1997), Monadic simultaneous rigid *E*-unification and related problems, *in* "Automata, Languages and Programming, 24th International Colloquium, ICALP'97" (P. Degano, R. Corrieri, and A. Marchetti-Spaccamella, Eds.), Lecture Notes in Computer Science, Vol. 1256, pp. 154–165, Springer Verlag.

Herbrand, J. (1972), "Logical Writings," Harvard Univ. Press, Cambridge, MA.

Hopcroft, J. E., and Ullman, J. D. (1979), "Introduction to Automata Theory, Languages and Computation," Addison–Wesley, Reading, MA.

Kozen, D. (1981), Positive first-order logic is NP-complete, *IBM J. Res. Develop.* **25**(4), 327–332.

Levy, J. (1998), Decidable and undecidable second-order unification problems, *in* "Rewriting Techniques and Applications, 9th International Conference, RTA-98, Tsukuba, Japan, March/April 1998, Proceedings" (T. Nipkow, Ed.), Vol. 1379, Lecture Notes in Computer Science, pp. 47–60, Springer-Verlag.

Levy, J., and Veanes, M. (1998), On unification problems in restricted second-order languages, *in* "Annual Conference of the European Association for Computer Science Logic (CSL'98), Brno, Czech Republic."

Lifschitz, V. (1967), Problem of decidability for some constructive theories of equalities, *Zap. Nauchn. Sem. LOMI* **4**. [Russian] English transl. *Sem. Math. Steklov Math. Inst.* **4**, Consultants Bureau, NY-London, 1969, pp. 29–31.

Makanin, G. (1977), The problem of solvability of equations in free semigroups, *Mat. Sb.* (*in Russian*) **103**(2), 147–236 [Russian]. English translation, *Amer. Math. Soc. Transl. (2)* **117**, 1981.

Mints, G. (1967), Choice of terms in quantifier rules of constructive predicate calculus (in Russian), *Zap. Nauchn. Sem. LOMI* **4**, 78–85. English Translation in: Seminars in Mathematics: Steklov Math. Inst. 4, Consultants Bureau, NY-London, 1969, pp. 43-46.

Orevkov, V. (1965), Unsolvability in the constructive predicate calculus of the class of the formulas of the type $\neg\neg\forall\exists$, *Soviet Math. Dokl.* **163**(3), 581–583. [in Russian]

Orevkov, V. (1976), Solvable classes of pseudo-prenex formulas (in Russian), *Zap. Nauchn. Sem. LOMI* **60**, 109–170. English translation in: Journal of Soviet Mathematics.

Plaisted, D. (1995), Special cases and substitutes for rigid *E*-unification, Technical Report MPI-I-95-2-010, Max-Planck-Institut für Informatik.

Snyder, W. (1980), Efficient ground completion: An $O(n \log n)$ algorithm for generating reduced sets of ground rewrite rules equivalent to a set of ground equations *E*, *in* "Rewriting Techniques and Applications" (G. Goos and J. Hartmanis, Eds.), Lecture Notes in Computer Science, Vol. 355, pp. 419–433, Springer-Verlag, New York/Berlin.

Statman, R. (1979), Lower bounds on Herbrand's theorem, *Proc. Amer. Math. Soc.* **75**(1), 104–107.

Veanes, M. (1996), "Uniform Representation of Recursively Enumerable Sets with Simultaneous Rigid *E*-Unification," UPMAIL Technical Report 126, Uppsala University Computing Science Department.

Veanes, M. (1997), The undecidability of simultaneous rigid *E*-unification with two variables, *in* "Proc. Kurt Gödel Colloquium KGC'97," Lecture Notes in Computer Science, Vol. 1289, 305–318, Springer-Verlag.

Veanes, M. (1998), The relation between second-order unification and simultaneous rigid *E*-unification, *in* "Proc. Thirteenth Annual IEEE Symposium on Logic in Computer Science, June 21-24, Indianapolis, Indiana (LICS'98)," pp. 264–275, IEEE Comput. Soc..

Voda, P., and Komara, J. (1995), "On Herbrand Skeletons," Technical report, Institute of Informatics Comenius University Bratislava [Revised January 1996].

Voronkov, A. (1996), Proof search in intuitionistic logic with equality, or back to simultaneous rigid *E*-unification, *in* "Automated Deduction-CADE-13" (M. McRobbie and J. Slaney, Eds.), Lecture Notes in Computer Science, Vol. 1104, pp. 32–46, New Brunswick, NJ, USA.

Voronkov, A. (1997), Strategies in rigid-variable methods, *in* "Proc. of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)" (M. Pollack, Ed.), Vol. 1, pp. 114–119, Nagoya, Japan.

Voronkov, A. (1998a), Herbrand's theorem, automated reasoning and semantic tableaux, *in* "Proc. Thirteenth Annual IEEE Symposium on Logic in Computer Science, June 21–24, 1998, Indianapolis, Indiana (LICS'98)," pp. 252–263, IEEE Comp. Soc..

Voronkov, A. (1998b), Simultaneous rigid *E*-unification and other decision problems related to Herbrand's theorem, *Theor. Comput. Sci.*, in press.